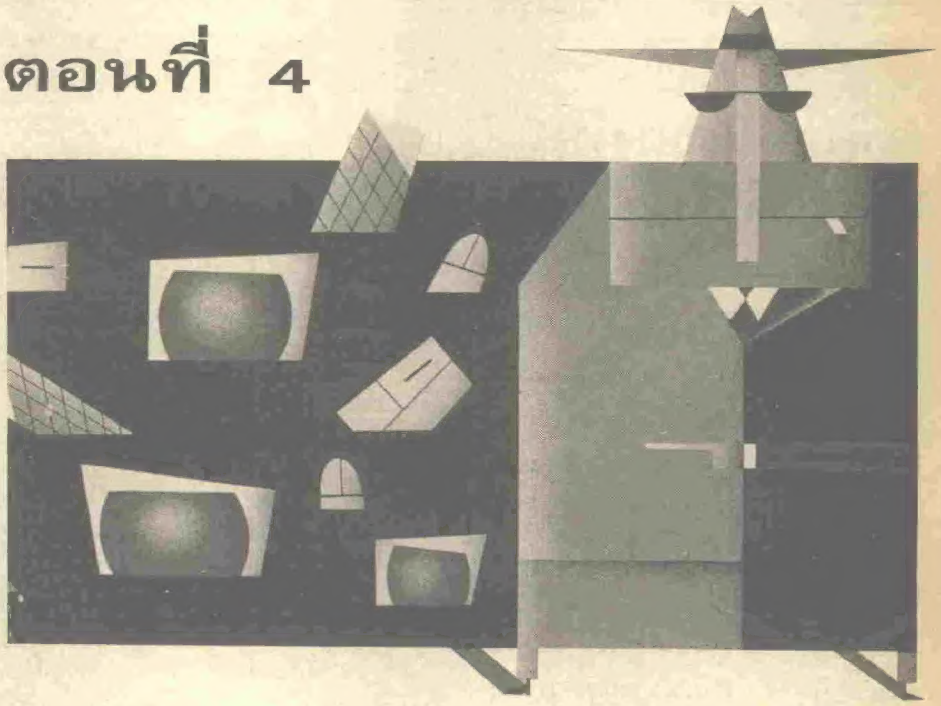
พงษ์ศักดิ์ หิรัณยรัชต์

ก้าวแรกในการสร้าง มาโครเอ็กเซล ตอนที่ 4

กล่าวถึงการสร้างฟังก์ชันที่ใช้ในการสร้างมาโคร พร้อมทั้งเครื่องมือที่ใช้ในการดีบักคำสั่งมาโคร



ในฉบับที่แล้วผู้เขียนได้กล่าวถึงฟังก์ชันที่ใช้ในการสร้างมาโครให้มีประสิทธิภาพและง่ายต่อการนำไปใช้ไปแล้ว ฉบับนี้จะเป็นเรื่องเกี่ยวกับการสร้างฟังก์ชัน ที่ใช้ในการสร้างมาโครและเครื่องมือในการดีบักคำสั่งมาโคร

การสร้างฟังก์ชัน

การสร้างฟังก์ชันในที่นี้หมายถึงการสร้างฟังก์ชันเพื่อใช้งานเฉพาะอย่าง โดยสร้างฟังก์ชันเก็บไว้ที่บนมาโครชีต ฟังก์ชันที่สร้างขึ้นนี้อาจประกอบด้วยฟังก์ชันมาตรฐาน (Build-in function) หรือสูตรการคำนวณ การใช้งานฟังก์ชันที่สร้างขึ้นมาเองนี้ก็เหมือนกับการใช้งานฟังก์ชันมาตรฐานทุกประการ

องค์ประกอบในการสร้าง

ในการสร้างฟังก์ชันเพื่อใช้งานเฉพาะอย่างซึ่งต่อไปนี้จะเรียกว่าการสร้างฟังก์ชัน มีองค์ประกอบอยู่ 3 ประการด้วยกันคือ

1. อาร์กิวเมนต์ของฟังก์ชัน
2. สูตรการคำนวณ
3. ค่าผลลัพธ์จากฟังก์ชัน

รายละเอียดขององค์ประกอบทั้ง 3 จะกล่าวต่อไปในบทความตอนนี้

ฟังก์ชันตัวอย่าง

เพื่อให้ผู้อ่านเห็นภาพขององค์ประกอบในการสร้างฟังก์ชัน จึงต้องสร้างฟังก์ชันขึ้นมาประกอบการอธิบาย ฟังก์ชันที่จะสร้างนี้เป็นฟังก์ชันง่าย ๆ ใช้ในการหาราคาขาย มีสูตรการคำนวณดังนี้

```
PRICE = COST + PROFIT
เมื่อ  PRICE  คือ ราคาขาย
       COST   คือ ต้นทุนสินค้า
       PROFIT คือ กำไร
```

รายละเอียดของฟังก์ชันแสดงในรูปที่ 1

	A	B	C	
1				
2		CalculatePrice	ฟังก์ชันคำนวณค่าราคาขาย	
3				
4		= RESULT(1)	กำหนดชนิดของข้อมูลให้กับผลลัพธ์	
5		= ARGUMENT("COST",1)	สร้างอาร์กิวเมนต์ชื่อ COST	
6		= ARGUMENT("PROFIT",1)	สร้างอาร์กิวเมนต์ชื่อ PROFIT	
7	PRICE	= COST + PROFIT	คำนวณหาค่าราคาขาย	
8		= RETURN(PRICE)	แสดงผลลัพธ์	
9				

รูปที่ 1 ฟังก์ชันตัวอย่าง CalculatePrice

ข้อพิจารณาในการสร้างอาร์กิวเมนต์ให้กับฟังก์ชัน

ในฟังก์ชันมาตรฐานก็มีการกำหนดอาร์กิวเมนต์เอาไว้ ฟังก์ชันที่สร้างขึ้นเองก็จำเป็นต้องมีการกำหนดอาร์กิวเมนต์ให้เช่นกัน อาร์กิวเมนต์คือ ค่าที่ใช้ในการคำนวณ เรากำหนดหรือสร้างอาร์กิวเมนต์ผ่านฟังก์ชัน ARGUMENT โดยที่ฟังก์ชัน ARGUMENT หนึ่ง ๆ สามารถกำหนดหรือสร้างอาร์กิวเมนต์เพื่อใช้งานได้เพียง 1 ตัวเท่านั้น อาร์กิวเมนต์ตัวแรกที่ผู้ใช้ฟังก์ชันป้อนเข้ามาคืออาร์กิวเมนต์ในฟังก์ชัน ARGUMENT บรรทัดแรกสุด และอาร์กิวเมนต์ตัวที่สองที่ผู้ใช้ป้อนเข้ามาคือ อาร์กิวเมนต์ในฟังก์ชัน ARGUMENT บรรทัดถัดมา กฎสำคัญในการใช้ฟังก์ชัน ARGUMENT คือ จะต้องระบุไว้ก่อนหน้าสูตรการคำนวณที่ใช้ค่าอาร์กิวเมนต์นั้น ๆ

รูปแบบของฟังก์ชัน ARGUMENT คือ

```
= ARGUMENT (ชื่ออาร์กิวเมนต์.รหัสชนิดของข้อมูล.เซลอ้างอิง)
```

ชื่ออาร์กิวเมนต์ จะต้องอยู่ในเครื่องหมาย " " เท่านั้น ชื่ออาร์กิวเมนต์ต้องไม่ไปซ้ำกับชื่อเซลที่อยู่ในโครชีต ถ้าในมาโครชีตมีเซลชื่อ MONDAY จะต้องไม่ตั้งชื่ออาร์กิวเมนต์เป็น MONDAY อย่างเด็ดขาด

รหัสชนิดของข้อมูล เพื่อเป็นการตรวจสอบชนิดของอาร์กิวเมนต์ที่ผู้ใช้ป้อนเข้ามาเป็นอาร์กิวเมนต์ เอ็กเซลจึงกำหนดให้มีการกำหนดชนิดของข้อมูลรายละเอียดดังแสดงในตารางที่ 1

อาร์กิวเมนต์ชนิดของข้อมูลสามารถเป็นได้ทั้งสูตรคำนวณ เซลอ้างอิง หรือค่าคงที่ (Constant) เพียงแต่ว่าผลลัพธ์สุดท้ายจะต้องเป็นตัวเลขซึ่งสอดคล้องกับรหัส

รหัสสามารถนำมารวมกันเพื่อสร้างความหลากหลายให้กับชนิดของข้อมูล เช่น ถ้าต้องการให้อาร์กิวเมนต์เป็นได้ทั้ง ตัวเลขและข้อความ ให้ระบุรหัสเป็น 3 (คือ ผลบวกของ 1 กับ 2) ถ้าไม่มีการระบุรหัสของข้อมูล เอ็กเซลจะกำหนดให้รหัสเป็น 7 โดยปริยาย ซึ่งหมายความว่า อาร์กิวเมนต์ในฟังก์ชัน ARGUMENT สามารถรับข้อมูลที่เป็นตัวเลขหรือข้อความ หรือตรรกก็ได้ (7 คือผลรวมของ 1+2+4)

ถ้าผู้ใช้ป้อนชนิดของข้อมูลไม่ตรงกับรหัสที่ระบุไว้ แรกสุดเอ็กเซลจะพยายามแปลงข้อมูล (Convert) ให้เป็นไปตามรหัสที่กำหนดไว้ แต่ถ้าการแปลงข้อมูลไม่ประสบผลสำเร็จ เอ็กเซลจะแสดงข้อความแสดงความผิดพลาดว่า # VALUE ที่เวอร์กชีตที่กำลังใช้งานฟังก์ชันนี้อยู่

เซลอ้างอิง คือ การระบุเซลหรือกลุ่มของเซลที่อยู่บนมาโครชีต

ข้อพิจารณาในการกำหนดค่าผลลัพธ์จากฟังก์ชัน

เราใช้ฟังก์ชัน RESULT ในการระบุชนิดของข้อมูลซึ่งเป็นผลลัพธ์จากฟังก์ชันที่สร้างขึ้น จะต้องระบุฟังก์ชัน

รหัส	ชนิดของข้อมูล
1	ตัวเลข ทำการบวก ลบ คูณ และ หารได้
2	ข้อความ
4	ตรรก
8	เซลอ้างอิง
16	Error
64	อะเรย์

ตารางที่ 1 รหัสและชนิดของข้อมูลของอาร์กิวเมนต์

RESULT ทุกครั้ง ถ้าต้องการผลลัพธ์ของฟังก์ชันที่สร้างขึ้นเป็นอะเรย์หรือเซลอ้างอิง

กฎเกณฑ์ในการใช้ฟังก์ชัน RESULT มีอยู่ว่า ฟังก์ชัน RESULT จะต้องระบุอยู่ก่อนหน้าสูตรการคำนวณหรือต้องอยู่ก่อนหน้าฟังก์ชัน ARGUMENT

รูปแบบของฟังก์ชันมีดังนี้

```
RESULT (รหัสชนิดของข้อมูล)
```

รหัสชนิดของข้อมูล คือ ชนิดข้อมูลซึ่งเป็นผลลัพธ์จากฟังก์ชันที่สร้างขึ้น รหัสและชนิดของข้อมูลใช้ร่วมกับชนิดของข้อมูลในฟังก์ชัน ARGUMENT ดังแสดงในตารางที่ 1 ถ้ารหัสชนิดของข้อมูลถูกละเอาไว้เอ็กเซลจะกำหนดให้มีรหัสเป็น 7 โดยปริยาย

ขั้นตอนในการสร้างฟังก์ชัน

การสร้างฟังก์ชันมีด้วยกัน 2 ขั้นตอนใหญ่ ๆ คือ

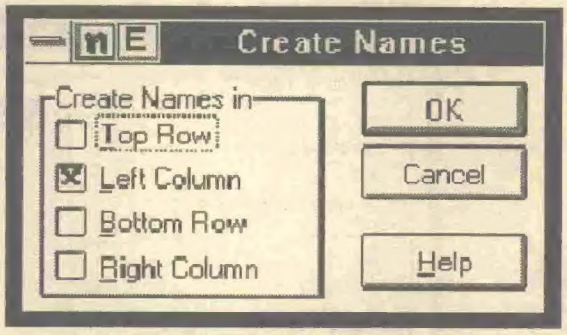
1. การสร้างสูตรการคำนวณบนมาโครชีต อาจสร้างบนชีตใหม่หรือชีตที่มีข้อมูลอยู่แล้วก็ได้
2. การกำหนดชื่อให้กับฟังก์ชัน

การทำงาน

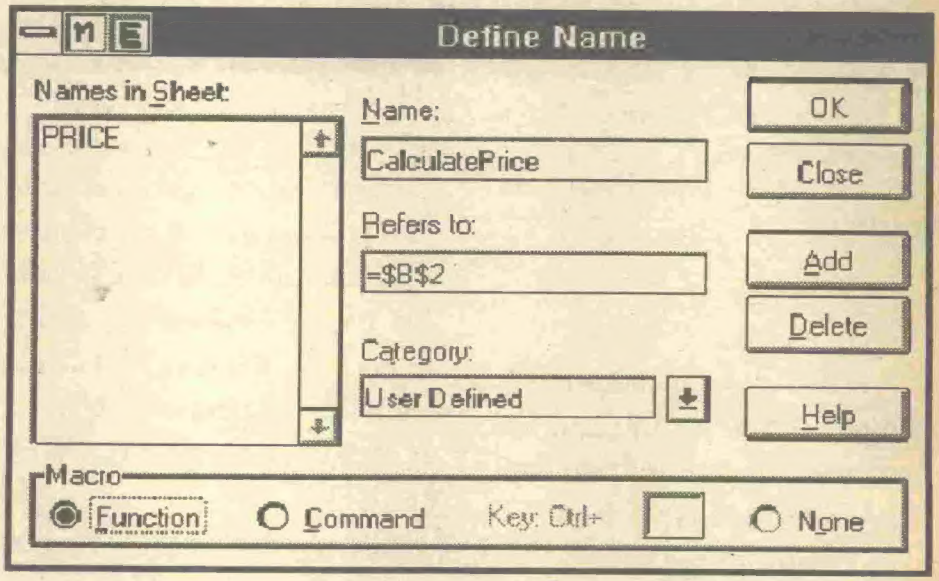
เอ็กเซลจะเริ่มต้นคำนวณสูตรการคำนวณในฟังก์ชันตั้งแต่บรรทัดแรกไปบรรทัดที่สองและสามในแนวคอลัมน์เรื่อยลงไปจนพบฟังก์ชัน RETURN จึงสิ้นสุดการทำงาน ดังนั้นจึงควรเขียนสูตรการคำนวณในแนวดิ่ง

ทดลองสร้างฟังก์ชัน

1. เปิดมาโครชีตอันใหม่ โดยไปที่เมนู File เลือก New แล้วเลือก Macro Sheet



รูปที่ 2
กรอบโต้ตอบ Create Names



รูปที่ 3 กรอบโต้ตอบ Define Name

2. ทำการป้อนรายละเอียดของฟังก์ชันตามรูปที่ 1

3. กำหนดชื่อให้เซล B7

3.1 เลือกเซล A7:B7 โดยการแดร็กเม้าส์ที่ไปที่เซลดังกล่าว

3.2 ไปที่เมนู Formula เลือก Create Names หน้าจอปรากฏกรอบโต้ตอบ Create Name ดังแสดงในรูปที่ 2 ให้คลิ๊กเม้าส์ที่ปุ่ม OK

4. กำหนดชื่อให้กับฟังก์ชัน

4.1 เลือกเซล B2 โดยการคลิ๊กเม้าส์ที่เซลดังกล่าว

4.2 ไปที่เมนู Formula เลือก Define Name

4.3 ที่หน้าจอจะปรากฏกรอบโต้ตอบ Define Name ดังแสดงในรูปที่ 3

4.4 ที่กรอบ Macro ให้คลิ๊กเม้าส์ที่หน้าหัวข้อ Function เพื่อเป็นการกำหนดให้ฟังก์ชันที่เรากำลังสร้างขึ้นนี้ (CalculatePrice) ปรากฏที่กรอบโต้ตอบ Paste Function เมื่อมีการเรียกใช้คำสั่ง Paste Function จากเมนู Formula

จะสังเกตเห็นว่าที่กรอบ Category ปรากฏข้อความว่า User Defined เป็นการระบุให้ฟังก์ชัน CalculatePrice เก็บอยู่ในหมวด (Category) User Defined เราสามารถกำหนดให้ฟังก์ชันที่

เราสร้างขึ้นเก็บอยู่ในหมวดใด ๆ ก็ได้ ในที่นี้ผู้เขียนต้องการเก็บไว้ในหมวด User Defined

5. ที่กรอบโต้ตอบ Define Name คลิ๊กเม้าส์ที่ปุ่ม OK

การใช้งานฟังก์ชันที่สร้างขึ้น

การใช้งานฟังก์ชันมี 2 วิธี

1. พิมพ์ผ่านแป้นพิมพ์ รูปแบบมีดังนี้

```
=ชื่อมาโครชีตเชื่อฟังก์ชันและอาร์กิวเมนต์
```

ตัวอย่างเช่น

```
=MACROS1.XLMICALCULATEPRICE
(500,200)
```

2. ใช้คำสั่ง Paste Function จากเมนู Formula

ขณะเรียกใช้งานฟังก์ชันมาโครชีตที่เก็บฟังก์ชันที่จะใช้งานต้องโหลดอยู่ในหน่วยความจำเสมอ ในที่นี้จะทดลองเรียกใช้งานฟังก์ชันด้วยวิธีที่ 2

1. ไปที่เวอร์กชีตใด ๆ ก็ได้

2. ที่เมนู Formula เลือก Paste Function แล้วเลือก User Defined ที่หน้าจอจะปรากฏกรอบโต้ตอบ Paste Function ดังแสดงในรูปที่ 4 ที่กรอบ Paste Function จะปรากฏชื่อฟังก์ชัน CalculatePrice

3. ให้คลิ๊กเม้าส์ที่ปุ่ม OK

4. ให้ป้อนค่าอาร์กิวเมนต์ดังนี้

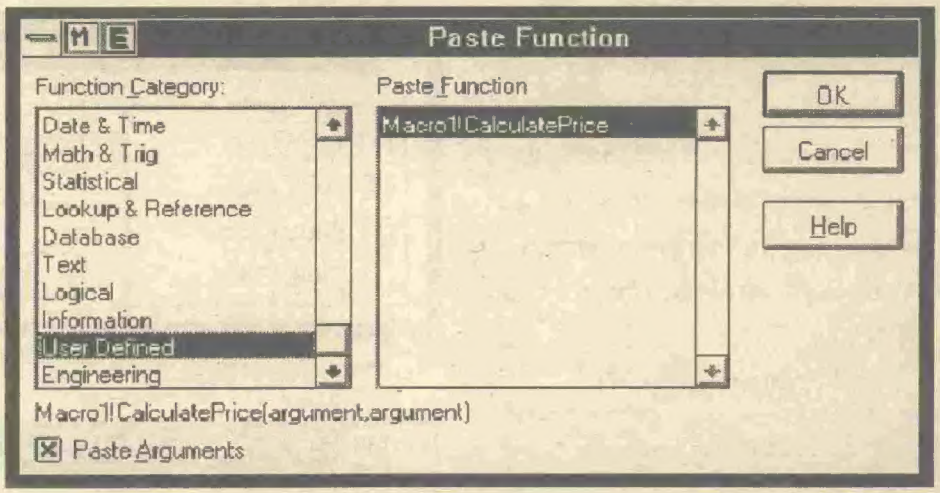
```
=Macro1!CalculatePrice(500,200)
```

5. เคาะแป้น Enter

6. ผลลัพธ์ที่ได้คือ 700

เครื่องมือในการดีบัก (Debug) คำสั่งมาโคร

สำหรับท่านที่มีประสบการณ์ด้านการเขียนโปรแกรมมาอย่างโชกโชนคงรู้จักคำว่าดีบักเป็นอย่างดี การตรวจสอบการทำงานของคำสั่งที่เราเขียนขึ้นว่าเป็นไปตามวัตถุประสงค์หรือไม่ หรือการตรวจสอบการทำงานของคำสั่ง เมื่อผลลัพธ์ได้ไม่ตรงกับความเป็นจริง การตรวจสอบการทำงานดังกล่าวเรียกว่าการดีบัก



รูปที่ 4 กรอบโต้ตอบ Paste Function

เอ็กเซลมีเครื่องมือในการตรวจสอบการดีบักอยู่ 2 อย่างคือ

1. การดีบักด้วยโหมด Single Step หมายถึงการรันคำสั่งมาโครทีละบรรทัดช้า ๆ

2. การดีบักด้วย มาโคร Add-in ชื่อ Debugger (มาโคร Add-in คือมาโครที่เอ็กเซลสร้างขึ้นมาให้ผู้ใช้ได้ใช้งานเพื่อความสะดวกในการทำงานบางประเภท ดูรายละเอียดที่หัวข้อมาโคร Add-in ในบทความนี้)

ในบทความนี้จะเสนอการดีบักด้วยโหมด Single Step แต่เพียงอย่างเดียว

ทูลบาร์มาโคร

ก่อนอธิบายการดีบักมาโครด้วย Single Step Mode จะขอแนะนำเครื่องมือที่มีประโยชน์ต่อการสร้างมาโคร เครื่องมือดังกล่าวคือ ทูลบาร์มาโคร (Macro toolbar)

ทูลบาร์มาโครประกอบไปด้วยเครื่องมือที่ใช้สำหรับการบันทึกการสร้างมาโคร การรันมาโคร การหยุดการทำงานของมาโครชั่วขณะ (Pause) และการตรวจสอบการทำงานของมาโครทีละคำสั่งจนสิ้นสุดคำสั่ง (Step through macros)

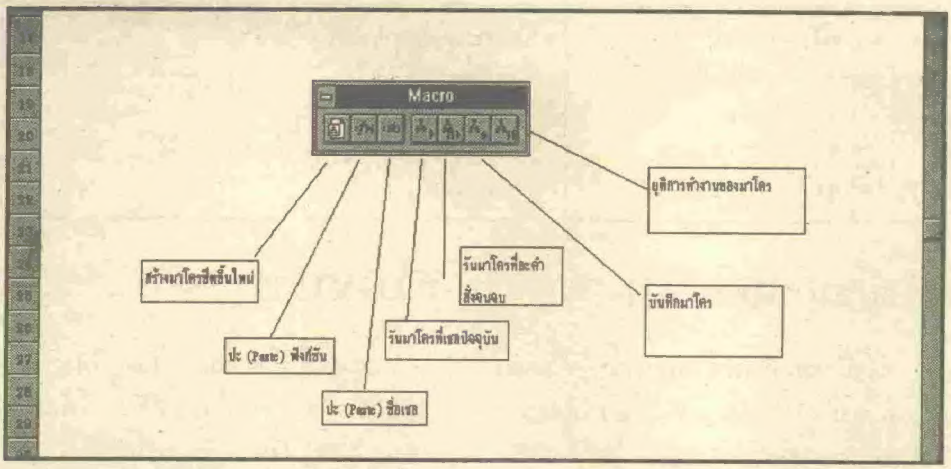
การเรียกใช้งานทูลบาร์มาโคร

1. คลิ๊กเม้าส์ที่ปุ่มขวาในบริเวณทูลบาร์มาตรฐาน (Standard toolbar)

2. เมื่อเมนู Shortcut ปรากฏให้คลิ๊กเม้าส์เลือก Macro ทูลบาร์มาโครจะปรากฏบนจอภาพ ดังแสดงในรูปที่ 5

เรียนรู้ฟังก์ชันเพิ่มเติม

ในการอธิบายการดีบักคำสั่งมาโครด้วยโหมด Single Step ให้เห็นภาพพจน์นั้น เห็นทีจะหลีกไม่พ้นการสร้างมาโครขึ้นมาเพื่อสาธิตการดีบัก เพื่อไม่ให้ผู้อ่านจำเจกับการสร้างมาโครด้วยฟังก์ชันซ้ำซาก ผู้เขียนจะขอแนะนำ



รูปที่ 5 ทูลบาร์มาโคร

ฟังก์ชันใหม่ที่เห็นว่ามีประโยชน์ต่อการใช้งานในอนาคต

เราจะสร้างมาโครชื่อ OnOffRecalc เพื่อใช้ประกอบการอธิบายการดีบักเป็นมาโครสำหรับตรวจสอบสถานภาพในการคำนวณซ้ำ (Recalculation) หลักการทำงานเป็นดังนี้ ทำการตรวจสอบสถานภาพการคำนวณซ้ำ ถ้าการคำนวณซ้ำเป็นแบบอัตโนมัติ (Automatic) ให้เปลี่ยนเป็นคำนวณซ้ำเมื่อสั่ง (Manual) และในทางกลับกัน ถ้าตรวจสอบสถานภาพการคำนวณซ้ำแล้วพบว่าเป็นแบบคำนวณซ้ำเมื่อสั่งให้ปรับเป็นแบบอัตโนมัติ จากนั้นให้แจ้งข้อความระบุชนิดของการคำนวณซ้ำที่หน้าจอภาพ ผังงานในรูปที่ 6 แสดงการทำงานของมาโครดังกล่าว

ฟังก์ชัน GET.DOCUMENT

เป็นฟังก์ชันใช้สำหรับตรวจสอบสถานภาพของเอกสาร (Document) ที่เราสนใจ หลังจากตรวจสอบแล้วฟังก์ชันจะให้ค่า (Return) ผลลัพธ์ที่ได้จากการตรวจสอบเพื่อใช้ประโยชน์ต่อไป

รูปแบบของฟังก์ชันแบบเป็นดังนี้

GET.DOCUMENT (รหัสประเภท, ชื่อเอกสาร)

รหัสประเภท คือรหัสประเภทของสถานภาพที่ต้องการตรวจสอบ รหัส 14 ใช้แทนการตรวจสอบสถานภาพการคำนวณซ้ำ

ชื่อเอกสาร คือชื่อแฟ้มเอกสารที่ต้องการตรวจสอบสถานภาพ หากละไว้หมายถึงแฟ้มที่เปิดใช้งานอยู่ในปัจจุบัน

ในที่นี้สิ่งที่เราต้องการตรวจสอบคือ สถานภาพการคำนวณซ้ำ มีรหัสประเภทเป็น 14 ดังนั้นจึงต้องระบุอาร์กิวเมนต์ให้กับฟังก์ชันดังนี้

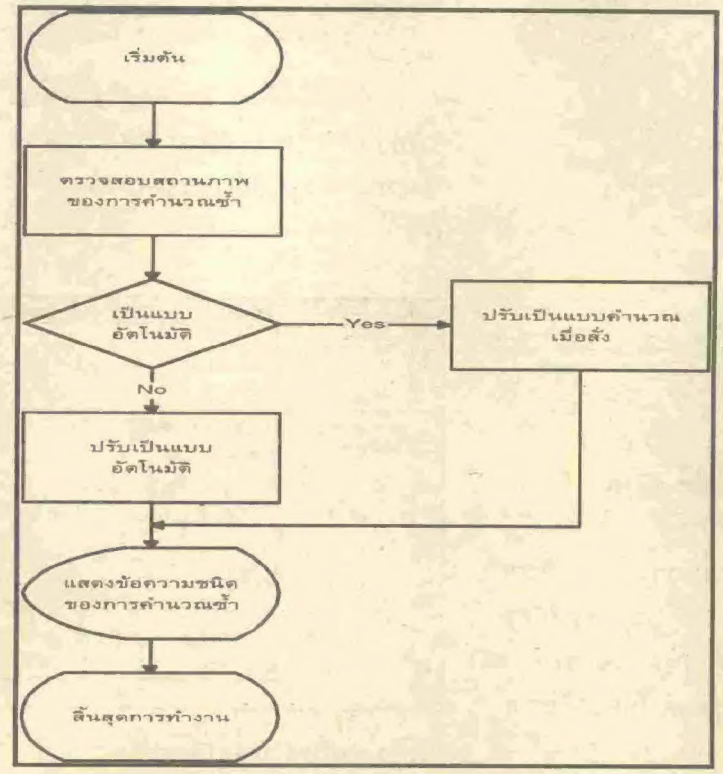
```
=GET.DOCUMENT(14)
```

ถ้าสถานภาพการคำนวณเป็นแบบอัตโนมัติ ฟังก์ชันจะให้ค่าผลลัพธ์เป็น 1 และถ้าสถานภาพเป็นคำนวณเมื่อสั่ง ฟังก์ชันจะให้ค่าผลลัพธ์เป็น 3

ฟังก์ชัน CALCULATION

เป็นฟังก์ชันที่ใช้ในการปรับค่าการคำนวณซ้ำเป็นแบบอัตโนมัติหรือแบบคำนวณซ้ำเมื่อสั่ง

รูปแบบของฟังก์ชันเป็นดังนี้



รูปที่ 6 ผังแสดงการทำงานของมาโคร OnOffRecalc

```
CALCULATION (รหัสการคำนวณซ้ำ)
Iter,max__nem,max__change,update,pre-
cision, date__1904,calc__save,save__values,
alt__exp,alt__form)
```

อาร์กิวเมนต์ของฟังก์ชันนี้ค่อนข้างยาวเป็นพิเศษ แต่ในที่นี้จะกล่าวเฉพาะอาร์กิวเมนต์ที่เราจะนำไปใช้ในมาโครที่จะสร้างขึ้นเท่านั้น นั่นคือรหัสการคำนวณซ้ำ

ความหมายของรหัสการคำนวณซ้ำดูได้จากตารางที่ 2

ตารางที่ 2 รหัสและความหมายของการคำนวณซ้ำ

รหัสการคำนวณซ้ำ	ชนิดของการคำนวณซ้ำ
1	อัตโนมัติ
2	อัตโนมัติยกเว้นข้อมูลในตาราง
3	ทำเมื่อสั่ง

ฟังก์ชัน IF...ELSE.IF... ELSE...END.IF

ในบทความก้าวแรกในการสร้างมาโคร เอ็กเซล ตอนที่ 2 ได้กล่าวถึงฟังก์ชัน IF ไปบ้างแล้ว แต่ยังไม่ครบรูปแบบของฟังก์ชัน IF ที่มีอยู่ ในการตรวจสอบสถานภาพของการคำนวณซ้ำเราจะใช้ฟังก์ชัน IF ทำการตรวจสอบ

รูปแบบของฟังก์ชันมีดังนี้

```
=IF (เงื่อนไข)
    คำสั่ง.....
    คำสั่ง.....
=ELSE.IF (เงื่อนไข)
    คำสั่ง.....
    คำสั่ง.....
=ELSE()
    คำสั่ง.....
    คำสั่ง.....
=END.IF()
```

การเขียนคำสั่งได้ IF และ ELSE ผู้เขียนขอแนะนำให้จัดย่อหน้าให้ตรงกันโดยการเคาะแป้นเว้นวรรคก่อนพิมพ์เครื่องหมายเท่ากับ (=) ประโยชน์ก็คือการตรวจทานความถูกต้องของคำสั่งทำได้สะดวกขึ้นไม่มีผลทำให้เวลาในการรันเร็วขึ้นแต่ประการใด

ฟังก์ชัน MESSAGE

เป็นฟังก์ชันที่ใช้สำหรับให้แสดงข้อความ (Message) ที่บรรทัดบอกสถานภาพ (Status bar)

รูปแบบของฟังก์ชันเป็นดังนี้

```
MESSAGE (เงื่อนไขตรรก, ข้อความ)
```

เงื่อนไขตรรก เป็นเงื่อนไขที่ใช้ในการควบคุมให้ข้อความปรากฏหรือไม่ปรากฏบนจอภาพ มี 2 เงื่อนไขคือ TRUE เป็นการบอกให้เอ็กเซลแสดงข้อความ และ FALSE เป็นการบอกให้เอ็กเซลลบข้อความที่ผู้ใช้สั่งให้แสดง ณ บรรทัดบอกสถานภาพ ก่อนพากลับคืนสู่สถานภาพปกติของบรรทัดบอกสถานภาพ

ข้อความ คือข้อความที่ผู้ใช้ต้องการให้ปรากฏที่บรรทัดบอกสถานภาพ ข้อความจะต้องอยู่ในเครื่องหมาย " " เท่านั้น

สร้างมาโคร

ตอนนี้ก็ได้เวลาสร้างมาโคร On-OffRecalc กันแล้วละครับ ให้สร้างมาโครตามรูปที่ 7 ขั้นตอนมีดังนี้

1. เปิดมาโครชีตขึ้นมาใหม่ ด้วยการไปที่เมนู File เลือก New จากนั้นเลือก Macro Sheet แล้วคลิ๊กปุ่ม OK
2. พิมพ์รายละเอียดตามรูปที่ 7
3. ตั้งชื่อเซล B4
   3.1 แดร็กเม้าส์เซล A4:B4
   3.2 ไปที่เมนู Formula เลือก Create Names
   3.3 ที่หน้าจอจะปรากฏกรอบโต้ตอบ Create Names ดังรูปที่ 2
   3.4 ให้คลิ๊กเม้าส์ที่ปุ่ม OK
4. ตั้งชื่อมาโคร
   4.1 คลิ๊กเม้าส์ที่ B2
   4.2 ไปที่เมนู Formula เลือก Define Name

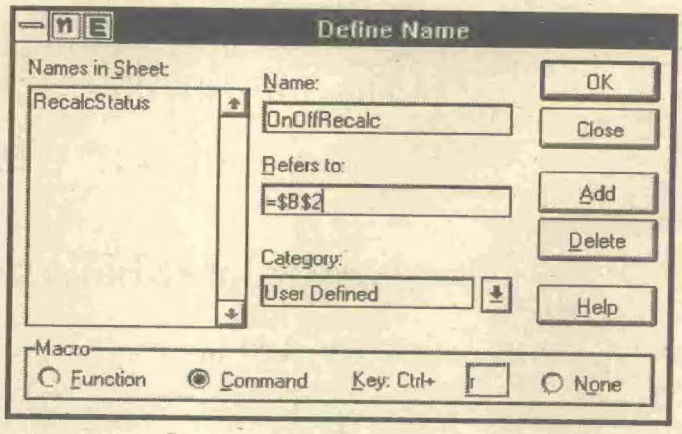
	A	B	C
1			
2	Ctrl + r	OnOffRecalc	
3			
4	RecalcStatus	=GET.DOCUMENT(14)	ตรวจสอบสถานภาพการคำนวณซ้ำ แล้วเก็บค่าผลลัพธ์ไว้
5		=IF(RecalcStatus=1)	ถ้าเป็นแบบอัตโนมัติ
6		= CALCULATION(3)	เปลี่ยนให้เป็นแบบทำเมื่อสั่ง
7		= MESSAGE(TRUE,"ขณะนี้การคำนวณซ้ำเป็นแบบทำเมื่อสั่ง")	
8		=ELSE()	ถ้าเป็นแบบทำเมื่อสั่ง
9		= CALCULATION(1)	เปลี่ยนให้เป็นแบบทำเมื่อสั่ง
10		= MESSAGE(TRUE,"ขณะนี้การคำนวณซ้ำเป็นแบบอัตโนมัติ")	
11		=END.IF()	
12		=RETURN()	สิ้นสุดการทำงาน
13			

รูปที่ 7 มาโคร OnOffRecalc

4.3 ที่หน้าจอจะปรากฏกรอบโต้ตอบ Define Name จัดการกำหนด ShortCut Key ตามรูปที่ 8

4.4 คลิ๊กปุ่ม OK

จากนั้นทดสอบการทำงานของมาโคร โดยไปที่เมนู Macro แล้วเลือก Run หรือใช้การกดแป้น Ctrl + r ก็ได้



รูปที่ 8 Define Name กรอบโต้ตอบ

วิธีการดีบักมาโครด้วยโหมด Single Step

การดีบักด้วยวิธีนี้เปรียบเสมือนการทำ Slow motion เราจะเห็นการทำงานของมาโครทีละคำสั่งอย่างช้า ๆ มีประโยชน์มากในการตรวจสอบหาข้อผิดพลาดที่ทำให้ผลลัพธ์ไม่เป็นไปตามต้องการ

การรันมาโคร OnOffRecalc ในโหมด Single Step

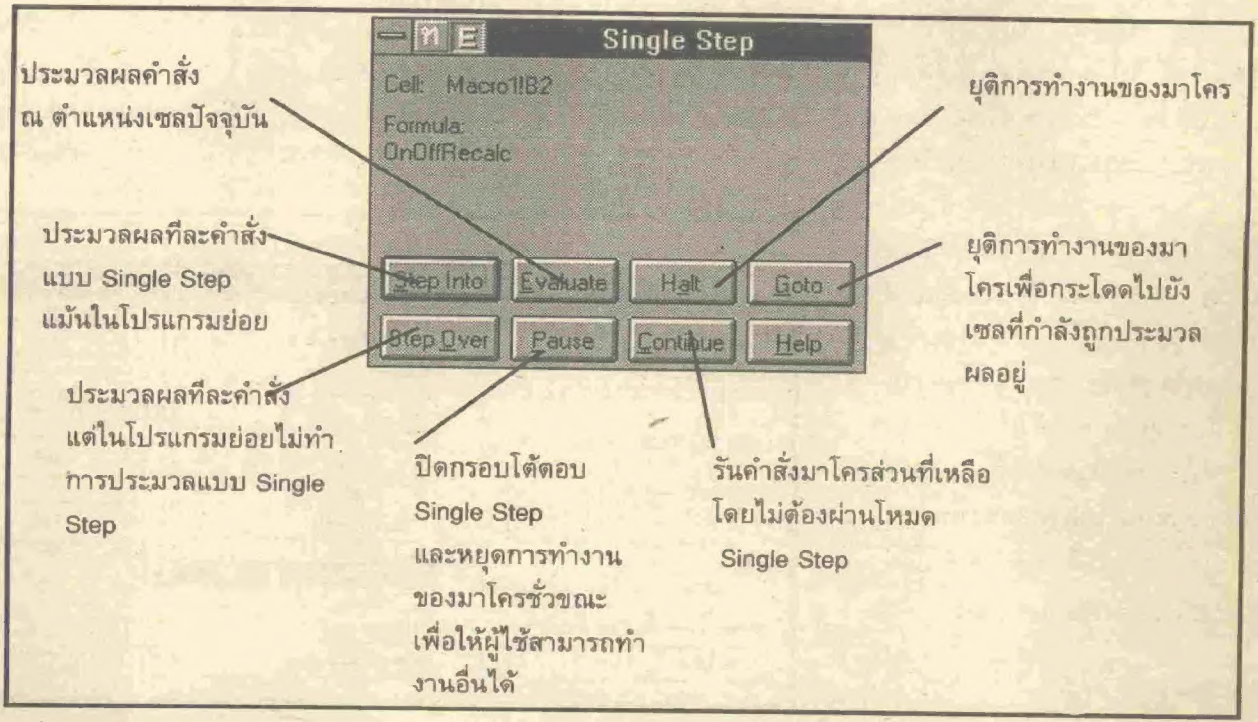
1. ไปที่มาโครชีตที่เก็บมาโคร OnOffRecalc แล้วคลิ๊กเม้าส์ที่ B2

2. ไปที่เมนู Macro เลือก Run กรอบโต้ตอบ Run จะปรากฏที่จอภาพ

3. ที่กรอบ Run เลือก OnOff-Recalc จากนั้นให้คลิ๊กเม้าส์ที่ปุ่ม Step ขณะนี้กรอบโต้ตอบ Single Step จะปรากฏบนจอดังรูปที่ 9

4. คลิ๊กปุ่ม Evaluate 2 ครั้ง ขณะนี้ในกรอบโต้ตอบ Single Step จะเห็นข้อความ =GET.DOCUMENT (14) ปรากฏใต้หัวข้อ Formula

5. คลิ๊กปุ่ม Evaluate อีกครั้ง คราวนี้ที่ใต้หัวข้อ Formula จะปรากฏข้อความเหมือนข้อความในข้อ 4 แตก-



รูปที่ 9 กรอบโต้ตอบ Single Step และคำอธิบาย

ต่างกันตรงที่ลักษณะของฟอนต์เท่านั้น เพราะฟอนต์ในขั้นตอนนี้เป็นตัวเอียง แสดงให้เห็นว่ากำลังประมวลคำสั่งบรรทัดนี้อยู่

6. คลิ๊กปุ่ม Evaluate อีกครั้ง จะมีเลข 3 ปรากฏขึ้น (ของผู้อ่านอาจเป็นเลข 1 ก็ได้ไม่ผิดกติกา) เลขที่ปรากฏนี้คือผลลัพธ์ที่ได้จากฟังก์ชัน =GET.DOCUMETN (14) นั่นเอง

7. คลิ๊กปุ่ม Evaluate อีกครั้งที่ได้หัวข้อ Formula จะแสดงผลที่ได้ (RecalcStatus = 1)

8. คลิ๊กปุ่ม Evaluate อีกครั้งที่ได้หัวข้อ Formula จะแสดงผลที่ได้จากการประมวลคำสั่งในข้อ 7 คือ =IF(FALSE)

9. คลิ๊กปุ่ม Halt เพื่อยุติการทำงานของมาโคร

มาโคร Add-in

มาโคร Add-in ก็เหมือนกับมาโครชีตทั่ว ๆ ไป แตกต่างกันที่ตรงมาโคร Add-in จะต้องเซฟไฟล์ให้อยู่ในฟอร์แมต Add-in เท่านั้น การเรียกใช้งานก็เหมือนกับการเรียกใช้งานมาโครชีตโดยทั่วไป กล่าวคือเลือกเมนู File แล้วเลือก Open แต่มาโคร Add-in มีข้อพิเศษอยู่ที่ขณะโหลดไฟล์ลงสู่หน่วยความจำ เอ็กเซลจะแสดงข้อความจำนวนเปอร์เซนต์ความสำเร็จในโหลดไฟล์ลงสู่หน่วยความจำที่มุมซ้ายบนของมาโครชีต และหลังจากโหลดลงสู่หน่วยความจำแล้วไฟล์จะไม่สามารถแก้ไขได้เหมือนกับมาโครชีตทั่ว ๆ ไป

☺ ☹

ข้อแตกต่างระหว่างมาโคร Add-in กับมาโครปกติ

มีข้อแตกต่างอยู่ด้วยกัน 3 ข้อ ดังในตารางที่ 3

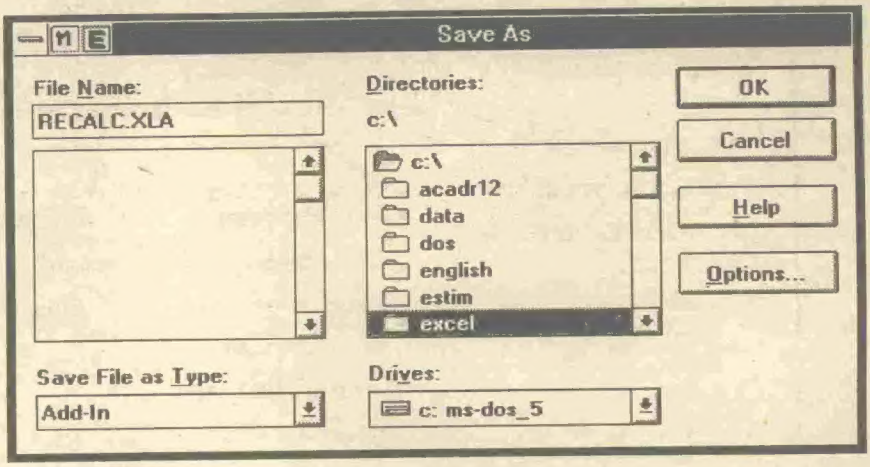
การเซฟมาโครชีตให้เป็นมาโคร Add-in

เราจะลองเซฟมาโครชีตที่เก็บมาโคร OnOffRecalc ให้เป็นมาโคร Add-in

1. ไปที่มาโครชีตที่เก็บมาโคร OnOffRecalc
2. ไปที่เมนู File เลือก Save As
3. ที่กรอบ File Name ให้ดับเบิ้ลคลิ๊กเม้าส์ แล้วพิมพ์ RECALC
4. ที่รายการ Save File as Type เลือก Add-In ดังรูปที่ 10
5. คลิ๊ก OK
6. ไปที่เมนู File เลือก Close เพื่อทำการปิดไฟล์

ตารางที่ 3 ตารางแสดงข้อแตกต่างระหว่างมาโคร Add-in กับมาโครปกติ

รายการ	มาโคร Add-in	มาโครปกติ
การแก้ไขหลังจากโหลดลงสู่หน่วยความจำ	ทำไม่ได้ เหตุผล : ป้องกันการแก้ไขด้วยความรู้เท่าไม่ถึงการหรือผู้สร้างมาโครไม่ต้องการให้ผู้อื่นทำการแก้ไขโดยพลการ	ทำได้
คำสั่งมาโครปรากฏในกรอบโต้ตอบ Run	ไม่ปรากฏ	ปรากฏ
การจัดเรียงฟังก์ชันในกรอบ Paste Function	จัดเรียงตามชื่อฟังก์ชันที่อยู่ในไฟล์มาโคร Add-in	ไม่จัดเรียงตามชื่อฟังก์ชันที่อยู่ในไฟล์มาโคร แต่จัดเรียงตามชื่อไฟล์



รูปที่ 10 กรอบโต้ตอบ Save As เพื่อเซฟมาโครแบบ Add-in

การโหลดมาโคร Add-in

1. ไปที่เมนู File เลือก Open
2. เลือกไฟล์ RECALC แล้วสังเกตที่มุมซ้ายบนของเวอร์กชีต

ทดลองกดแป้น Ctrl+r สลับไปมาจะเห็นข้อความสลับกันไปมาที่บรรทัดบอกสถานภาพ ข้อสังเกตอีกประการหนึ่งขณะทำการโหลดไฟล์ RECALC ก็คือไฟล์มีขนาดเล็กข้อความที่บอกเปอร์เซนต์ความสำเร็จ ในการโหลดไฟล์จึงมองไม่เห็น

การแก้ไขมาโคร Add-in

ผู้อ่านคงสงสัยในใจว่า ก็ไหนบอกว่าไม่สามารถแก้ไขไฟล์มาโคร Add-in ได้อย่างไงล่ะ? จริง ๆ แล้วแก้ไขได้ครับ ขั้นตอนมีดังนี้

เมื่อมาโคร Add-in ได้โหลดลงสู่หน่วยความจำแล้ว

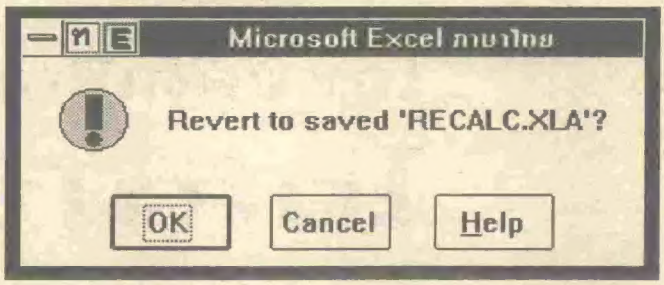
1. ไปที่เมนู File เลือก Open
2. ขณะนี้ให้กดแป้น Shift ค้างเอาไว้ แล้วเลือกไฟล์ RECALC จากนั้นคลิ๊กปุ่ม OK หน้าจอจะเป็นดังรูปที่ 11
3. คลิ๊กปุ่ม OK
4. ไปที่เมนู Window เลือก Unhide
5. ที่หน้าจอจะปรากฏกรอบโต้ตอบ Unhide ให้เลือก RECALC.XLA

สรุป

ในการคำนวณเพื่อให้ได้ผลลัพธ์ที่ต้องการผ่านคำสั่งมาโคร ถ้าข้อมูลดิบที่รับเข้ามาประมวลผลมีจำนวนไม่มากจนเกินไป ควรสร้างมาโครนั้นให้เป็นฟังก์ชัน แล้ววางแผนเปลี่ยนวิธีการรับข้อมูลดิบจากผู้ใช้ให้กลายมาเป็นการป้อนอาร์กิวเมนต์ของฟังก์ชันแทน

จุดเด่นอีกประการหนึ่งของเอ็กเซลในการสร้างคำมาโครก็คือมีเครื่องมือในการดีบักคำสั่ง ทำให้หาข้อผิดพลาดในการทำงานได้ง่ายขึ้นและเร็วขึ้น เมื่อมีข้อผิดพลาดในการทำงานของคำสั่งมาโครหรือต้องการค้นหาสาเหตุที่ทำให้ผลลัพธ์ที่ได้ไม่ตรงตามต้องการ ควรดีบักคำสั่งด้วยโหมด Single Step

มาโคร Add-in เหมาะสำหรับงานที่ไม่ต้องการผู้ใช้ในระดับเริ่มต้นป้องกันการแก้ไขคำสั่งมาโครโดยไม่เจตนา



รูปที่ 11
กรอบ Revert

ก่อนจาก

เมื่อวันที่ 22 มีนาคม ที่ผ่านมา ผู้เขียนมีโอกาสเข้าร่วมสัมมนาการเปิดตัวอย่างเป็นทางการของไมโครซอฟต์ เอ็กเซล 5.0 ไทย เอดิชัน มีสิ่งแปลกใหม่ที่แตกต่างจากเอ็กเซล 4.0 ซึ่งน่าสนใจ ผู้เขียนจะนำมาเสนอให้ทราบในโอกาสต่อไป